UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/769,535	01/30/2004	Milton E. Moskowitz	H0005134- -1623	8642

128 7590 08/04/2009

HONEYWELL INTERNATIONAL INC.
PATENT SERVICES
101 COLUMBIA ROAD
P O BOX 2245
MORRISTOWN, NJ 07962-2245

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
08/04/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/769,535	MOSKOWITZ ET AL.	
	Examiner	Art Unit	
	TUAN A. VU	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 June 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18,21 and 22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18, 21-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 7/14/09.

As indicated in Applicant's response, claims 1-18, 21-22 have been amended, and claims 19-20 canceled. Claims 1-18, 21-22 are pending in the office action.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-18, 21-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Charisius et al, USPN: 6,983,446 (hereinafter Charisius)

As per claim 1, Charisius discloses a method for verifying first generated computer code having a plurality of lines (e.g. 2102 Fig. 21; 1302 – Fig. 13)generated by a code generating module from a model file of a system including a plurality of functions generated by a model module, the method comprising:

processing the model file (Fig. 13, 15-16, 21-23), by a verification module, to determine values, inputs, outputs (e.g. *Number of Members* – Table 1, *return a value* – Table 11; attributes – Table 12), function type, and syntax for the plurality of functions (e.g. metrics 1900, 1906 – Fig. 19A, B; Note: audit directives 2110-2112, 2114, 2116, 2118, 2120 – see Fig. 21 to 23 – reads on processing model file – see Fig. 23 -- to derive metrics – see Table 1 to Table 18 col. 10-20); and

generating second computer code (Fig. 8A; 806 – Fig. 8B; 812 – Fig. 8C; Fig. 19A – Note: computer code as tips, audit acronym/semantics for selection and user guidance and code reference for correction guidance – Fig. 8C, 19C -- reads on second computer code) for the model file based on the determined values, inputs, outputs, function type, and syntax (see Table 1 to Table 18 col. 10-20)

comparing each line of the first generated computer code and the second code generated by the verification module (code 2102 – Fig. 21; step 2002, Fig. 20A) to determine if the first generated computer code includes correct values, correct inputs, correct outputs, correct functions, and correct syntax (e.g. Fig. 4; Fig. 19B-C; Fig. 8B-C; code 1302 – Fig. 13 – Note: audit module to validate first generated code or non-validated source code - Fig. 3, col. 6, lines 12-22 - against requirements and language specifications or audit tips – see Fig. 8A; 806 – Fig. 8B; 812 – Fig. 8C; Fig. 19A; 2110-2112, 2114, 2116, 2118, 2120, Fig. 21 - that encompass rules definitions supported by audit metrics **reads on** comparing lines of first code and second code and, the second code conveyed as audit/verification requirements including underlying support from metrics as correct values, input/outputs, functions syntax – see Table 1, Table 3, Table 4 from above); and

transmitting an error message if one a line of the first generated computer code does not include a correct syntax based on the comparison (e.g. error message … conform to predefined styles; error message when an audit is performed – col. 18 lines 50-55; col. 19 lines 28-34, 52-55; error message - col. 20 lines 25-67)

Charisius does not explicitly disclose transmitting the error message *if one or more lines of first generated code does not include a correct value, correct input, correct output, a correct*

function. Using a QA module operating on predefined audit error types as set forth above (Fig. 3; Fig. 13-14; col. 5, lines 61-64; col. 6, lines 12-22; *definitions, templates*, Fig. 8B-C; col. 7, lines 63 to col. 8, line 21; Fig. 19A, 20A –col. 16, lines 9-14 – and/or metrics as correct values, input/outputs, functions syntax – see Table 1, Table 3, Table 4 from above;), Charisius discloses error message to be shown to developers regarding many respects, such as syntax, format parameters of a function, declaring of class or members and this is indicative that any violation determined by a verification or audit process to input/output declarations, or correct values or functions would be raised by the validation process. Programming language syntax in the nature of input/output requirements, type conformance, value conformance would have been evident or suggested from the above Charisius' presentation of audit rules or metrics and code conformance guidance; while display of error in a validation endeavor such as Charisius's auditing tool such as a visual indication to user about effects from the verifying was a well-known concept as shown above. For one of ordinary skill in the art and based on Charisius's validating tool according to well known practice (in software development) of using Charisius' tool to systematically notify the developers using visual means on any propriety issue regarding code form or implementation thereof, it would have been obvious for one skill in the art at the time the invention was made to implement the error messages by the source code auditing in Charisius so that error message would be displayed when auditing a line of expected source if the line does not include specifically a correct value, correct input, correct output, a correct function. One of ordinary skill in the art would be motivated to do so because not having such proper value or function, correct input or output thereof a code might not properly compile notably when such

type of deficiencies would necessarily fall under the types or style of errors contemplated by a auditing tool based on the above.

As per claim 2, Charisius teaches comparing each line of the first generated computer code and the second generated code to determine if the first generated computer code is missing a line of code (is missing – col 15 lines 47-48; col 25 lines 55-59); and, by virtue of the rationale in claim 1, transmitting the error message if the if the first generated computer code is missing the line of code.

As per claim 3, Charisius teaches comparing each line of the first generated computer code and the second generated code to determine if the first generated computer code includes any an extraneous line of code (col. 20 Table 18; col. 22 lines 61-67); and by virtue of the rationale in claim 1, transmitting the error message if the the generated computer code includes the extraneous line of code.

As per claim 4, Charisius discloses comparing each line of the generated computer code and the second generated code to determine if the [the] generated computer code is in a logical order (ordered properly – col. 33, lines 55-60 ; col. 34 lines 10-44) and by virtue of the rationale in claim 1, transmitting the error message if the first generated computer code is not in the logical order.

As per claim 5, Charisius discloses comparing a first header information section in the first generated computer code and second header information section in the second generated code to determine if the first header information section matches (*Declaration* – cols. 25-26; match a declaration, col. 37, lines 1-37- Note: generated source code or class package declaration with respect to expected declaration in OO class or Use case package – see Fig. 14-15, 22 -- in

an *audit* instance reads on comparing header of a class signature declaration – Note: second header information treated as any form of expected header – see USC 112, 1st para) the second header information section; and (re rationale in claim 1) transmitting the error message if the first header information section does not match the second header information section.

As per claim 6, Charisius discloses comparing a first declared variable section in the first generated computer code and a second declared variable section in the expected code (col. 29 lines 7 to col. 30 lines 67; col. 25 lines 20-52 – Note: fixed source code based on positions of declarations, declared type of member with respect to their signature and constructor with respect to a parent class reads on comparing generated variable section with variable section of expected code) to determine if the first declared variable section matches the second declared variable section; and (refer to rationale in claim 1) transmitting the error message if the first declared variable section does not match the second declared variable section.

As per claim 7, Charisius discloses a computer-readable storage medium containing a set of instructions for verifying a first generated computer code (refer to claim 1) having a plurality of lines from a code generating module, the generated computer code automatically generated from a model file of a system having a plurality of functions and created by a model module, the set of instructions comprising:

code that reads in the model file; code that determines values, inputs, outputs, function type, and syntax for each of the plurality of functions in the model file (refer to claim 1);

code that generates a second generated computer code based on the determined values, inputs, outputs, function type, and syntax; code that reads in the first generated computer code (refer to claim 1);

code that compares each line in the first generated computer code and the second generated computer code to determine if the first generated computer code includes the determined values, inputs, outputs, function type, and syntax in the second generated computer code (refer to claim 1); and

code that transmits an error message if a first line in the first generated computer code does not include a determined syntax based on the comparison (refer to claim 1).

Charisius does not explicitly disclose transmitting the error message *if one or more lines of first generated code does not include a correct value, correct input, correct output, a correct function*. However, this limitation has been addressed as obvious in claim 1.

As per claim 8, Charisius discloses wherein the set of instructions further comprises:

code that compares each line in the first generated computer code and the second generated computer code (refer to claim 7); and

code that transmits the error message if the first line does not include the determined value, the determined input, the determined output, the determined function, the determined syntax, or combinations thereof (refer to rationale in claim 1).

As per claim 9, refer to the rationale of claim 2 (line not determined of claim 9 treated as missing line),

As per claims 10-11, refer to claims 4-5

As per claim 12, Charisius discloses a system for verifying the contents of a first generated computer code having a plurality of lines generated by a code (refer to first computer

code in claim 1) generating module from a model file including a plurality of functions generated by a model module, comprising:

a processor operable to:

process the model file to determine values, inputs, outputs, function type, and syntax for each of the plurality of functions (refer to claim 1) and generate a second generated computer code (refer to claim 1) for the model file based on the determined values, inputs, outputs, functions type, and syntax for the code model file,

compare each line in the first generated computer code with the second generated computer code to determine if the first generated computer code includes correct values, correct inputs, correct outputs, correct functions, and correct syntax (refer to claim 1), and

transmit an error message if a line in the first generated computer code does not include a correct syntax based on the comparison (refer to claim 1); and a display configured to display the error message, the display coupled to the processor (see error message).

Charisius does not explicitly disclose transmitting the error message if one or more lines of first generated code does not include a correct value, correct input, correct output, a correct function. However, this limitation has been addressed as obvious in claim 1

As per claim 13, refer to claim 2.

As per claim 14, Charisius discloses (by virtue of the obviousness in claim 12) wherein the error message indicates if a line of the first generated computer code has any additional content (e.g. *True ... False literals should not be used ... amount of meaningless code ... use of type casts not necessary ... 'abstract' considered obsolete ... should not use parentheses* – col. 19 lines 47 to col. 20 line 23; col. 16 line 58 to col. 17, line 37).

As per claim 15, Charisius discloses wherein the processor is operable to compare each line in the second generated computer code to the first generated computer code to determine if the plurality of lines are in an expected form, and transmit the error message if one or more of the lines of code in the plurality of lines do not match the expected form (e.g. error message … conform to predefined styles; error message when an audit is performed – col. 18 lines 50-55; col. 19 lines 28-34, 52-55; error message - col. 20 lines 25-67).

As per claims 16-18, refer to claim 3-5, respectively.

As per claim 21, Charisius discloses method of claim 1, further comprising the steps of: comparing each line in the first generated computer code and the second generated computer code to determine if the plurality of lines are complete(col. 16 line 55 to col. 17 line 37; avoid empty catch block … without empty body – Table 17 col. 18-19); and transmitting an error message if the plurality of lines are incomplete (refer to obviousness rationale in claim 1).

As per claim 22, Charisius discloses method of claim 7, wherein the set of instructions further comprises code that compares each line in the first generated computer code to the second generated computer code to determine if the first generated computer code is complete (refer to claim 21); and code that transmits the error message if the first generated computer code is incomplete.

Response to Arguments

4. Applicant's arguments filed 7/14/09 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

(A) Applicants have submitted that (Appl. Rmrks pg. 11) source code display being dynamically update in Charisius's graphical tool does not teach generating two codes generated

from a same model and used to verify values, inputs/outputs, function type, syntax. First, the code syntax by itself would be broad to include syntax regarding proper values, function type, input/output requirements for such function; hence all of input/output, type or values to verify by a audit or verification module would fall under such the ambit of verify syntax and would render the specific nature as to what is verified obvious, be it value, input, or data type. Second, the claim language as amended now recites 'second generated computer code', and does not specify exactly what this (semantically or syntactically) amounts to; nor does the claim dictate the scenario and code generating steps in terms as to how first code is contextually generated, (then displayed for comparison) such that a second code subsequent to the first generating, is generated in turn and amounts to either an internal or external expression (or graphical representation) that does have a particular format or syntax being distinct from the first code or similar to it, both taking as input the exact model file in a two distinct stages and yielding for each stage a particular form of generated code. The argument is also deemed not commensurate with the office Action that was on record for addressing a previous language of the claim prior to this latest amendment. In all, the language of 'second generated code' is not compelling specific to obviate the cited portions in Charisius from being applied; nor is it proper for the argument to raise a deficiency on the part of a previous Office Action that, in its own time context, did and was never meant to address merits of the currently claimed 'second generated code'. The argument is moot or largely non-commensurate with the Office action of record; not to mention that the broad language of 'second code' will be addressed based on broad reasonable interpretation, in view of the lack of specific teaching as analyzed from above.

(B) Applicants have submitted that level of ordinary skill in the art would not suffice to overcome the deficiencies of Charisius since the ordinary skill in the art (OSA) does not provide teachings (Appl. Rmrks pg. 12) in terms that it can be applied in the Office Action (as to complement Charisius), in order to fulfill 'comparing each line of the generated computer code and the expected code to determine … correct syntax'. OSA has been used to put together the teachings by Charisius which is display of errors message in a program validating process where metrics and requirements about constructing class, method, the propriety about adding members to classes, and OO code form/syntax are clearly taught in the Table 1-11 by the audit software. The 'expected code' and line has been addressed with a lack of support as per a USC 112 Rejection, which is not a part of the current Office Action. The current Rejection has been necessitated by the Amendment; and the above argument is deemed largely moot, or at least amounts to an empty allegation that does not contain evidence showing where in the 103 rationale is there a flaw that cannot render a particular feature claimed obvious.

(C) Applicants have submitted that for claims 2-6, 21, 7-11, 22, expected code based on the determined values, inputs, outputs function type and syntax as well as the comparing of two code cannot be fulfilled as per Charisius and OSA(Appl. Rmrks pg. 12 bottom, pg. 13 top). These remarks are not accompanied with facts demonstrating where in the cited parts of Charisius and the rationale for obviousness, some features are either missing or counter-teaching the very meaning or concept of the claim. The arguments are not persuasive.

(D) The arguments submitted regarding 'expected computer code … based on determined values, inputs … syntax' (Appl. Rmrks pg. 13) for claims 12-18, amount to a rehash of how the

office fail to provide obviousness by way of Charisius and OSA; hence will be referred back to section B, C from above.

In all, the claims have been interpreted based on the language currently submitted and will stand rejected until proper arguments would be sufficient in showing how exactly Charisius distinguish over that language (notwithstanding broad reasonable interpretation of one skill in the art)

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.
Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

August 01, 2009